```java
public void getMaxProfit() {

        // sort identifiers based on the the v/w of the items they refer to
        List<Integer> identifierList = Arrays.stream(identifiers)
                .boxed()
                .collect(Collectors.toList());
        @SuppressWarnings("unchecked")
        ArrayList<Integer> sortedIdentifers = new ArrayList(identifierList);
        Collections.sort(sortedIdentifers, (right, left) ->
Double.compare(vWRatio[identifierList.indexOf((left))],
                vWRatio[identifierList.indexOf((right))]));
        int tempCapacity = 0, removed = -1, tempValue = 0;

        int[] pickedUpItems = new int[n]; // to store picked up items,
initialized to 0s by jvm

        // keep picking up the highest v/w item until we run out of capacity
        while (tempCapacity < capacity && !sortedIdentifers.isEmpty()) {
            removed = sortedIdentifers.remove(0);
            if (tempCapacity + weights[removed - 1] <= capacity) {
                pickedUpItems[removed - 1] = 1;
                tempValue += values[removed - 1];
                tempCapacity += weights[removed - 1];
            }
        }
    }
```